

# A Minimum Jerk Predictor for Teleoperation with Variable Time Delay

Christian Smith\* and Henrik I Christensen†

\* Centre for Autonomous Systems, Royal Institute of Technology  
Stockholm, Sweden. Email: ccs@kth.se

†Robotics and Intelligent Machines, Georgia Institute of Technology  
Atlanta, GA 30332-0760. Email: hic@cc.gatech.edu

**Abstract**—In this paper we describe a method for bridging internet time delays in a teleoperation scenario. In the scenario, the sizes of the time delays is not only stochastic, but it is also large compared to the task execution time. The method proposed uses minimum jerk motion models to predict the input from the user a time into the future that is equivalent to the one-way communication delay. We present results from a teleoperated ball-catching experiment with real internet delays, where we show that the proposed method makes a significant improvement over traditional methods for teleoperation over intercontinental distances.

## I. INTRODUCTION

The desire to transfer human action over space, time and scale is well-known by now. One of the most widely advertised examples is the NASA Mars Rover Program. Other uses of teleoperation include medical robotics, hazardous material handling and teleoperated drones for military surveillance.

Teleoperation has been widely used in a variety of applications and is in many respects considered a mature technology, but few studies have considered teleoperation in the presence of significant dynamics [1]–[4]. The qualifier “significant” dynamics is used here to specify a process where the time-delay is a large fraction of the process cycle time.

A time-delay challenges control as it introduces a phase shift that make the control more difficult. For small time-delays it is easy to model it explicitly. For applications with time delays longer than the process time, such as the Mars rover program or control of some systems on-board the international space station or the space shuttle, the adopted strategy has been to use task oriented control [1], [2], [5].

A model frequently used with shorter time delays is to introduce a Smith predictor for compensation for the system delay [6], [7]. The Smith predictor inherently limits bandwidth of the overall system and in addition it is only well suited for systems with deterministic delays. Wave variable approaches have been successful in producing stable bilateral teleoperation with force feedback [8]. These have been extended with predictive systems with varying time-delays for handling stochastic channels, as in [9]. However, the bandwidth of interaction is still limited, leaving highly dynamic tasks untractable.

To understand this problem we have chosen to study the challenge of teleoperated catching. When catching an object thrown across a room, the flight time is on the order of 600–900 ms, and for intercontinental teleoperation across

the internet typical roundtrip delays are in the range of 100–200 ms for normal socket operations, which is on the order of 15–25% of the overall flying time. How does one select a control strategy for such systems?

It is well known that humans are visually dominated [10], and there are clear models that suggest that much of human actions are feed-forward driven rather than feedback driven [11]. Under such a control regime one would expect that motion for interaction is largely prospective, in particular for dynamic situations. If a person were to catch a thrown object directly then s/he would perform a ballistic arm movement followed by some correction [12], [13]. Given such a strategy, how can the early motion by the user be estimated and predicted with sufficient accuracy to allow it be to used to generate an early motion control signal for the end-effector? In previous work, it has been shown that *minimum jerk* motion models can be used to predict the input of a human operator [14], [15]. The novel contribution of the present paper is integrating these predictions into a closed loop teleoperation control system with time delays, and verifying the performance in experiments.

We study the estimation of commands issued by the user under significant time pressure. The questions addressed are:

- What are good estimation models for identification of the trajectory specified by the user?
- How can these models be integrated into the control system?
- What kind of performance can be achieved with a real system?

It is worth noting that others have studied the design of ball-catching strategies earlier. Autonomous robotic ball-catching has been described since the 1980’s [16]–[19]. Human ball-catching has also been widely studied [20], [21]. At DLR, human catching has been analyzed with the aim of improving trajectory generation for robotic catching [22], and at NTUA human motion models have been employed in order to use EMG to control robot catching for 1-dimensional trajectories [23]. However, teleoperated catching with free-flying 3-dimensional trajectories is a novel problem, to the best knowledge of the authors. Minimum Jerk models have also been applied in real time to human input in other work, with the aim of generating virtual fixtures that help an operator to avoid collisions while maintaining as much transparency as possible [24].

The paper is organized as follows: Section II describes the

teleoperated ball-catching task and the associated problems, Section III describes the solution approach using user input prediction, Section IV describes the time delayed teleoperation experiment used for evaluation and presents its results, and Section V concludes the paper.

## II. TASK DESCRIPTION

Ballcatching has been chosen as an example task in order to study teleoperation of a process where the time-delay is a large fraction of the process cycle time. The task for the operator is to guide the robot's end effector so that it intercepts the ballistic trajectory of the ball. The operator has to perform this without any direct observations of the ball or the manipulator, with all information being relayed through a user interface.

Given the specifications of the available experimental setup, the task can be specified more exactly as catching a ball that has been thrown a distance of 4 meters for a flight time of approximately 900 ms. The ball is caught in a cylinder if the spatial precision of the catch is within 4 cm. A detailed description of the manipulator and the catching task can be found in [25].

As an illustration of the difficulties that arise with delays, the setup was teleoperated with the user being located in Pontedera, Italy, and the manipulator being in Stockholm, Sweden. The average round-trip delay when using a public internet connection was measured to 50 ms. In that test, one single ball was caught, possibly the first successful trans-continental teleoperated ball-catching experiment of its kind. However, an examination of the subsequent misses showed that in most cases a correct intercept position was reached, but too late to be in time to catch the ball. Thus it is of interest to find a technique to bridge the time-delay problem. The current paper aims to find a means to do this so that a human operator can successfully perform a dynamically challenging task even in the presence of these significant delays.

## III. IMPLEMENTATION

### A. Control Strategy for Teleoperation

A classical control theory approach to coping with fixed time delays is the Smith predictor [6], [7]. More recently, time delays in bilateral teleoperation have been successfully dealt with using wave variables [8]. Combining this approach with Smith predictors and Kalman filtering, it has also been successfully applied to stochastic time delays [9]. One main reason for the success of wave variables is that they force passivity, and thereby stability, onto the time delayed teleoperation setup. However, this stability comes at the cost of responsiveness, and it has been shown that the performance degrades significantly as the magnitude of the delay approaches the time constants of the control signal [26]. In the scenario described here, the order of magnitude of the time constants is around 100 ms.

Another reason for the success of wave variables is their applicability to bilateral force-feedback control. In our ball-catching scenario, force feedback is not thought to be of

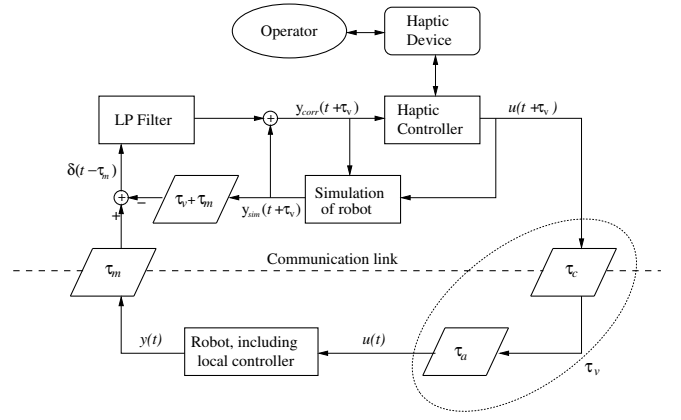


Fig. 1. Control structure A, with Smith predictor only (skewed boxes represent delays).

importance, as the manipulator will only move freely through space without contact with any other objects than the balls.

Therefore, we select a slightly different approach. We apply both Smith predictor and Kalman filtering as in [9], but do not include the wave variable transform. With the Smith predictor approach, the remote slave site is simulated locally at the master site, and measurements are used to correct the simulation to prevent it from diverging from the actual state of the remote site. In the employed scheme, everything displayed to the operator through the user interface is generated interacting exclusively with the dynamic simulation running on the operator interface computer. Since this interaction does not involve a time delay, the risk of instabilities is significantly reduced. This limits the approach to tasks where all significant aspects of the environment can be modeled with sufficient accuracy.

To close the control loop over the robot via the communication link, we can use a non-linear, multivariate Smith predictor control structure (depicted in Figure 1). The details of this implementation can be found in [27].

The command and measurement communication delays,  $\tau_c$  and  $\tau_m$  respectively, are handled in two parts. The first part handles the stochastic aspect of the delay by adding an artificial delay  $\tau_a$  to the command delay  $\tau_c$  when a command packet arrives at the robot so that their sum, the virtual delay  $\tau_v$ , is constant, as described in [27], [28]. In effect, the stochastic delay is exchanged for a slightly longer deterministic delay. Since the internet time delay is stochastic, it is expected that a small portion of packets will be delayed more than  $\tau_v$ . These are treated as dropped packets and ignored. Since the average time delay may vary over time,  $\tau_v$  is continuously recalculated. The allowed rate of change is very slow compared to other time constants in the system, typically  $\tau_v$  can change with a millisecond or less over several hundred packets.

The simulation result  $y_{sim}(t + \tau_v)$  is then delayed by  $\tau_v + \tau_m$  before it is compared to the measured state  $y(t - \tau_m)$  to form the simulation correction  $\delta$ . This means that when a measurement packet arrives from the robot, the current value

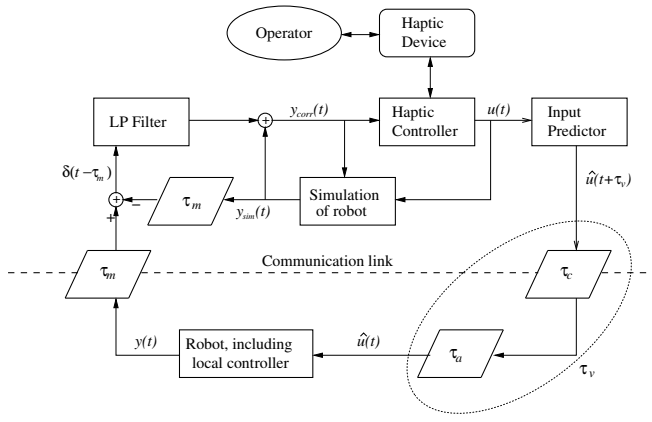


Fig. 2. Control structure B, with Smith predictor and input predictor (skewed boxes represent delays)

of  $\tau_m$  is calculated from timestamps and an old simulation result  $y_{sim}(t - \tau_m)$  retrieved from memory for comparison.

The net effect of this compensation is that the simulation, the haptic controller, and the operator all perceive time  $\tau_v$  ahead of real time. That allows the generated command signal  $u$  to travel to the robot before it is needed by the robot controller. Values of  $\tau_v$  upto 200 ms have been tested successfully.

The correction signal  $\delta$  is low pass filtered to reduce zero order hold noise that would otherwise be felt by the operator as vibrations of the same frequency as the communication link, 100 Hz.

The largest drawback with this approach is the need to simulate the entire roundtrip delay, including the artificial padding. The time difference between the simulation and the measurements is thereby  $\tau_v + \tau_m$ . This does not pose a large problem for aspects of the remote site that are easy to simulate, such as the robot dynamics or ball in ballistic flight. However, it is problematic when unexpected events occur. For instance, there is no advance warning for when a new ball is launched, and the trajectory can therefore not be simulated until the ball has already been in flight for a time corresponding to  $\tau_v + \tau_m$ . Thus, the operator's reactions will be delayed by the entire roundtrip delay.

We therefore propose a slightly different control structure. Instead of handling the entire roundtrip delay when predicting the remote state with a simulation, the delay handling is split into two separate predictors. The simulation of the remote site needs only bridge the shorter delay  $\tau_m$ , while the delay in the outgoing signal  $\tau_v$  is handled by predicting the future control signal,  $\hat{u}(t + \tau_v)$ . The principal structure for this approach is shown in Figure 2. With this approach, the operator's reaction to an unexpected event will only be delayed by the one-way delay  $\tau_m$ . The design of the input predictor is described in the following section.

## B. Simulation Models

In the present setup, models of the manipulator and ball path are well-known and simple enough to simulate within

small error margins. This only solves one part of the time-delay problem. The operator will have access to a good simulation of real-time information, but there will still be delays as the user reacts to this information, and as the system relays user input to the remote manipulator. In the present paper, the first steps towards including a simulation of the operator's input in the system are taken. This requires a model of the user's input that relates previous and present inputs to future ones.

A well-known model for explaining the kinematics of visually guided human reaching motions is the *minimum jerk* (MJ) model. It was first proposed for single-joint motions in [29], and later extended to include multi-joint planar motion in [30]. It was observed that the trajectory of voluntary arm motions, when described in extra-corporal cartesian space, follow certain constraints. The trajectories can be predicted by using a model in which the square sum of the third derivative of position, *jerk*, integrated over time is minimized. I.e., given a starting point, an end point and a time to move between the two, the trajectory that minimizes the jerk on this interval is the MJ trajectory. Observations in [13] and [22] on the motions that humans make when freely catching a thrown ball indicate that they start by moving towards the expected point of impact with a distinct MJ-type reaching motion, and later add smaller corrective MJ-type motions to accurately catch the ball.

More recently, other, more detailed models like minimum joint torque change, minimum force, or minimum energy have been proposed and shown to describe human motions more accurately [12], [13]. However, the more detailed models require detailed knowledge of posture and intrinsic mechanical parameters of the human subject, parameters that are not observable when using a hand-held haptic device for motion input. On the other hand, the MJ model can be completely described in cartesian space coordinates with no explicit knowledge of the subject's intrinsic mechanical properties, and is thus possible to implement in our scenario. Furthermore, when a subject is using the haptic device it is reasonable to assume that no external forces other than gravity act on the user, and that posture does not change significantly during the motion. In this case, it has been shown that trajectories predicted by the MJ model do not differ significantly from ones predicted by the more advanced models [12].

All MJ trajectories share the property that the 6th derivative is zero for the duration of the motion, and that they thus can be described as 5th degree polynomials, as in Equation 1.

$$\mathbf{x}(t) = a_1 t^5 + a_2 t^4 + a_3 t^3 + a_4 t^2 + a_5 t + a_6 \quad (1)$$

If we also add the start and end points of the motion,  $\mathbf{x}(t_0)$  and  $\mathbf{x}(t_1)$ , and state the position, velocity, and acceleration at these points, we get the following constraints on Equation 1.

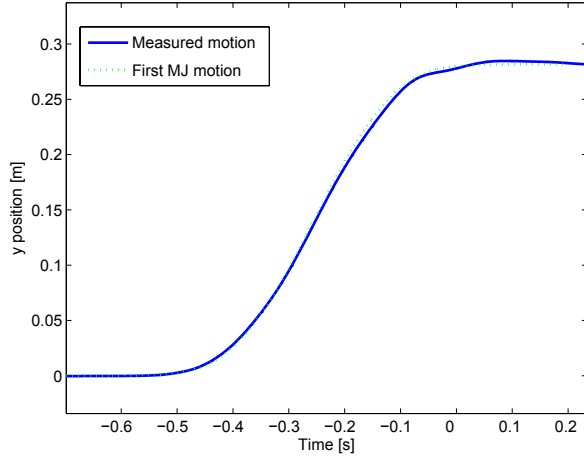


Fig. 3. Y component of measured hand trajectory with MJ trajectory fitted. In this case the hand trajectory contains only one major MJ component.

$$\begin{aligned} \mathbf{x}(t_0) &= \mathbf{x}_0, & \mathbf{x}(t_1) &= \mathbf{x}_1 \\ \dot{\mathbf{x}}(t_0) &= \dot{\mathbf{x}}_0, & \dot{\mathbf{x}}(t_1) &= \dot{\mathbf{x}}_1 \\ \ddot{\mathbf{x}}(t_0) &= \ddot{\mathbf{x}}_0, & \ddot{\mathbf{x}}(t_1) &= \ddot{\mathbf{x}}_1 \end{aligned}$$

The above constraints will give us 6 equations, and we get a well-defined system to find the 6 parameters  $a_1 \dots a_6$ . Thus, there is only one possible MJ trajectory for a given start and end, and it can be found by solving a simple system of linear equations.

The trajectories described by the MJ model are limited to one single motion. What happens if a more complex motion is desired or if the target of the motion is changed in mid-motion can be described by superpositioning several MJ trajectories. If the added MJ trajectory has an initial position, velocity, and acceleration of zero, this will still result in a continuous motion where the 6th derivative is zero, so the jerk is still minimized. This is described thoroughly in [31] and [32].

### C. Human Input Estimation

The MJ trajectory model described in Section III-B will be used to predict future user inputs. To predict future motions, it is necessary to know when the motion starts and stops. Given that the goal is predicting the outcome of a motion, it is plausible to register the starting point of the motion after it starts, but the point in time at which the motion ends has to be found before it can be observed. Using Equation 1, we see that if the start of a motion is known, but not the end, there are two unknowns — the time and position at which the motion ends. Thus, in theory, if two points on the trajectory are known, the system of equations can be solved to find these two unknowns and thereby specify the entire motion exactly. Since this involves solving a 5th degree polynomial, the solution will be very sensitive to error, especially in the time domain. However, if the start and end times are known, extrapolations that fit well with observed data are possible.

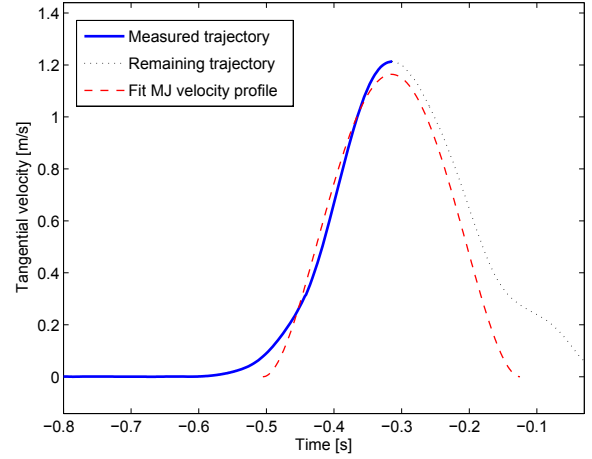


Fig. 4. 4th degree MJ velocity profile fitted to Kalman filtered velocity data. The solid line is the data used for fitting, the dotted line is the (unused) remainder of the measured velocity, and the dashed line represents the fit MJ velocity profile.

A robust way to detect the start time  $t_0$  and the end time  $t_1$  is proposed in the following. Since the motion itself is expected to follow a 5th degree polynomial curve, the velocity profile is expected to follow a 4th degree polynomial curve that starts and ends at zero value, with zero first derivative. Such a curve is symmetric around the apex. Empiric analysis of recorded motion data shows that by using an extended Kalman filter, the point of maximum velocity  $t_{v_{max}}$  is easy to detect [15]. Using a least square approach, the 4th degree velocity profile can be fit around this peak, fitting only to data before the peak, see Fig 4. The zeros of the polynomial found with this approach are used as candidates for the start and end of the motion, and used when fitting the motion data to a MJ trajectory.

A weakness in this approach is that half of the motion has to be observed before a prediction can be made. In order to facilitate earlier predictions, the EKF observer can be used to predict when  $t_{v_{max}}$  will be reached. This involves estimating higher derivatives, and is prone to high uncertainties in the presence of observation noise. In practice this means that a stable estimate of when  $t_{v_{max}}$  will be reached can be achieved after approximately one third of the motion. The implementation used in the experiments described later in this paper therefore uses a  $t_{v_{max}}$  predicted from the EKF until  $t_{v_{max}}$  is reached, when the algorithm switches to using the observed value.

When  $t_0$  and  $t_1$  are known, along with the start position  $x_0$  of the motion, Equation 1 can be rewritten as in 2.

$$\begin{aligned} \mathbf{x}(t) &= \\ &\mathbf{x}(t_0) + \\ &a_1 \left( t^5 - \frac{5}{2}(t_1 + t_0)t^4 + \frac{5}{3}(t_1^2 + 4t_1 t_0 + t_0^2)t^3 - \right. \\ &\quad \left. 5(t_1 t_0^2 - t_1^2 t_0)t^2 + 5t_0^2 t_1^2 t - \right. \\ &\quad \left. \frac{1}{6}t_0^5 + \frac{5}{6}t_0^4 t_1 - \frac{5}{3}t_0^3 t_1^2 \right) \end{aligned} \quad (2)$$

This equation only contains one unknown,  $a_1$ . By fitting to the latest measured data points with a least squares fitter, a fairly robust trajectory prediction is possible. This approach has been shown to predict accurately for delays up to approximately 70 ms [14]. This is comparable to the one-way internet delay between Stockholm, Sweden and Atlanta, Georgia, which has been measured to average 63 ms.

#### D. Robot Platform

The robot used in this experiment is a fast and lightweight manipulator with six degrees of freedom, see Figure 5. The robot can reach any point in its 60 cm  $\times$  60 cm workspace within 550 ms if it is started in a centered position, with the settings used in this experiment.

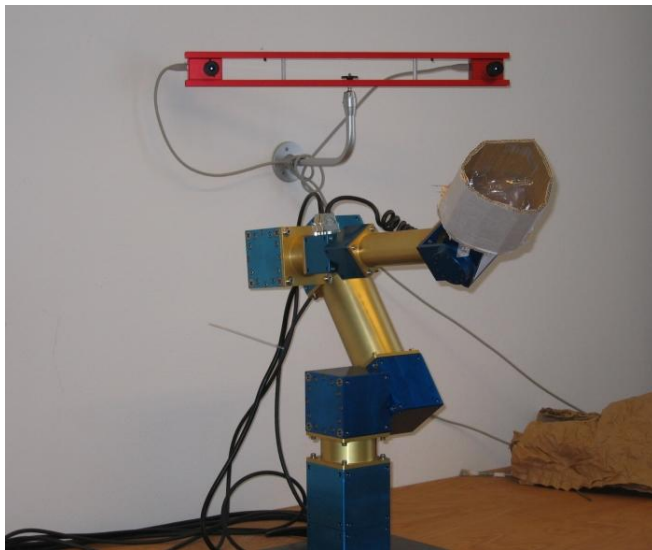


Fig. 5. The manipulator with cameras and ball-catching end effector.

The end effector used is a simple cardboard cylinder with a 7 cm radius. This means that the 7 cm diameter balls used in this experiment are caught if the center of mass hits within 4 cm of the center of the cylinder, as all hits to the rim tend to bounce off.

The system uses a 60 cm baseline stereo camera pair at 50 Hz to track thrown balls. Using a Kalman filter, this setup can predict the future trajectory of the ball within a few centimeters after approximately 5 images, or 100 ms. A detailed description of the manipulator and of the control setup and the camera tracking system has been presented in earlier work [15], [25], [27].

#### E. User Interface

The user interface (shown in Figure 6) used for the experiment consists of a stereo display realized with a 21-inch CRT monitor and shutter glasses, connected to a computer with a professional grade graphics board. An Omega haptic unit from Force Dimension served as input channel for user hand motion. This haptic device uses parallel linkage, and provides high stiffness and force output in a large workspace. A loudspeaker was used for audio feedback.



Fig. 6. User interface hardware configuration, with a CRT stereo display, an Omega haptic device, and a loudspeaker.

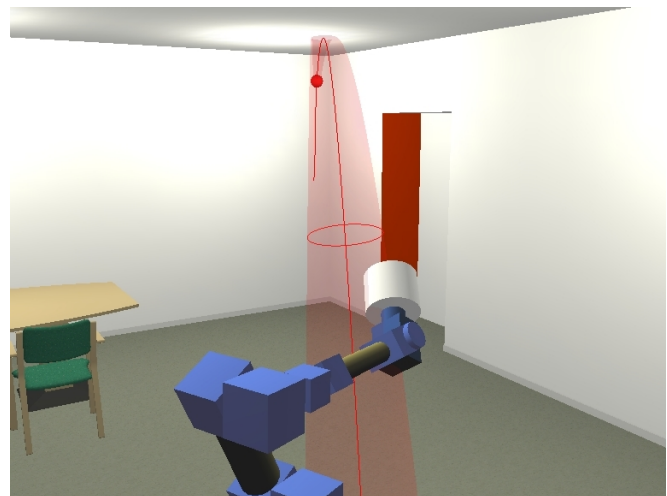


Fig. 7. The user's view in the ball-catching experiment.

The user interface receives estimated positions of the ball from the stereo vision system of the physical robot setup. It also gets the complete state of the vision system Kalman filter to be able to estimate future ball positions, which are used to show the projected trajectory of the ball as a red line through the ball, with the uncertainty inherent in the future Kalman states displayed as a semi-transparent shell at a distance of one standard deviation from the predicted trajectory. A predicted ball trajectory and its uncertainty representation are shown in Figure 7.

When a throw started, the word 'go' was output to the speaker to alert the user. Another sound was played when the ball was caught. The user interface is described in detail in [15].

## IV. EVALUATION EXPERIMENTS

The MJ based predictor was evaluated by an experiment using a ball-catching task that requires fast execution, while all signals were subject to significant time delays. For comparison, a Smith predictor setup, as detailed in



Section III-A was also evaluated in the same setting. In the following description, the traditional Smith predictor is called *system A*, and the Smith predictor augmented with the MJ input predictor is called *system B*, c.f. Figures 1 and 2.

#### A. Experimental Procedure

10 Novice subjects were asked to catch thrown balls using the teleoperation setup. The subjects were student volunteers, all male, ages from 23 to 30. They had no prior experience with the setup, and were not told any details of the control systems prior to the experiments. The subjects were given a brief instruction to the user interface and the ball-catching task, and were allowed to get acquainted with the setup by practicing catching balls until they felt comfortable with the task. This took between 30 and 50 tries for most subjects.

The performance in the presence of time delays was tested using actual internet time delays, for maximum realism. For practical reasons, both the slave and master sides were located in the same laboratory, but all communication signals between the two were delayed according to actual internet delays (including completely dropped packets) sampled during 24 hours starting from 11:40 GMT, Sep 25 2008.

One sample set consisted of delays measured between the Royal Institute of Technology (KTH) in Stockholm, Sweden and the University of Genua (UGE) in Italy. The median one-way delay time for these packets was 26.6 ms, and the standard deviation was 59.1 ms. The other sample set was generated by timing packet travel times from KTH to Georgia Institute of Technology (GT), in Atlanta, Georgia, USA. These packets had a median travel time of 63.0 ms, but with a much lower standard deviation of 0.2 ms. A deterministic one-way time delay of approximately 2.5 milliseconds was added by the rest of the communication loop.

In the experiment, for each subject 5 different settings were tried, with 20 throws per setting. The ordering of the settings were randomly permuted between subjects. The settings were:

- No delay, used for comparison.
- System A, UGE delays
- System A, GT delays
- System B, UGE delays
- System B, GT delays

To ensure repeatability of the experiment, the balls were launched using a mechanical launcher with a precision of  $\pm 10$  cm for this distance (see Fig 8). The balls were sent according to a pregenerated random pattern that was the same for all trials.

Earlier studies have shown that teleoperated ball-catching is a difficult task, and that high failure rates discourage subjects from performing their best [15]. Therefore the task was made slightly easier overall by adding a corrective action to the robot controller. This tries to center the end effector on the predicted trajectory, if the setpoint for the end effector comes within 5 cm of the predicted ball trajectory. This means that the subject only has to come within 5 cm of the ball trajectory in order to catch successfully, instead of 4 cm, and the effective target area is increased by 56%.



Fig. 8. The mechanical ball launcher

TABLE I  
RESULTS FROM BALL-CATCHING EXPERIMENT

| Setup               | Success rate ( $\pm 2\sigma$ ) |
|---------------------|--------------------------------|
| No delay            | 0.288 ( $\pm 0.045$ )          |
| System A, UGE delay | 0.256 ( $\pm 0.062$ )          |
| System A, GT delay  | 0.160 ( $\pm 0.052$ )          |
| System B, UGE delay | 0.275 ( $\pm 0.063$ )          |
| System B, GT delay  | 0.280 ( $\pm 0.065$ )          |

#### B. Results

The performance results are presented in Table I. With the longer GT delay times, the performance with System A drops to approximately half the success rate that was achieved with zero delays, while the success rate of system B drops by less than 0.01 compared to the zero delay baseline. This result is statistically significant at  $p < 0.05$ .

With the shorter UGE delays, the success rate of System A drops by 0.03, while the success rate of System B drops by 0.01 as compared to the zero delay baseline. However, here the difference in performance is not large enough to be statistically significant.

The average virtual delay  $\tau_v$  used by the system was 31.8 ms for the University of Genua delays, and 69.3 ms for the Georgia Tech delays.

These results show that the MJ predictor approach can successfully bridge time delays of more than 60 ms, while a traditional Smith predictor approach encounters problems. A probable cause for this is that, with the Smith predictor, transient events such as the firing of a ball with the ball launcher will be delayed the entire roundtrip time, or approximately 130 ms in the case of GT delays. Given that the subject only has a few hundred milliseconds to react, this delay is too large for successful task completion. On the

other hand, with the MJ approach, the transients will only be delayed by half as much, giving the user just enough time to react. As shown previously, user input can be predicted successfully up to 70 ms, so this is close to the limit of what the MJ predictor approach would be expected to cope with. In geographic terms, the difference of the two approaches is if a dynamic internet-based teleoperation task is limited to the same continent (within Europe), or if it is feasible to perform it intercontinentally (between Europe and North America).

## V. CONCLUSIONS

When time delays in a teleoperation setting begin to approach the time constants associated with performing the desired task, it is difficult to bridge the delays with traditional approaches. As shown in the experiment in the present paper, a setup with a minimum jerk based input predictor outperforms a standard Smith predictor with local simulation when the roundtrip delays approach 130 ms.

There are some limitations to the proposed approach. In its current implementation, it does not allow for sensory exploration of an unknown remote environment, as all relevant aspects of the remote site must be simulated locally. Also, the MJ model is limited to free motions, and is not suited for contact interaction, where significant external forces act on the manipulator.

## REFERENCES

- [1] G. Hirzinger, K. Landzettel, and C. Fagerer, "Telerobotics with large time delays," in *Proc. of the Intl. Conf. on Intelligent Robots and Systems*, 1994, pp. 571–578.
- [2] M. R. Stein and R. P. Paul, "Operator interaction, for time-delayed teleoperation, with a behavior-based controller," in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 1. IEEE, 1994, pp. 231–236.
- [3] G. Niemeyer and J.-J. Slotine, "Designing force reflecting teleoperators with large time delays to appear as virtual tools," in *Robotics and Automation*, 1997. *Proceedings.*, 1997 IEEE International Conference on, vol. 3, Apr 1997, pp. 2212–2218.
- [4] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikawa, "Ground-space bilateral teleoperation of ets-vii robot arm by direct bilateral coupling under 7-s time delay condition," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 3, pp. 499–511, June 2004.
- [5] B. K. Muirhead, "Mars rovers, past and future," in *Proceedings of the 2004 IEEE Aerospace Conference*, vol. 1, 2004, pp. 128–134.
- [6] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. Boston, MA: Addison-Wesley, 1995.
- [7] O. J. M. Smith, "A controller to overcome dead time," *ISA Journal*, vol. 6, pp. 28–33, Feb 1959.
- [8] G. Niemeyer and J.-J. Slotine, "Using wave variables for system analysis and robot control," in *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*. Albuquerque, NM: IEEE, Apr 1997, pp. 1619–1625.
- [9] S. Munir and W. J. Book, "Internet-based teleoperation using wave variables with prediction," *IEEE/ASME Transactions on Mechatronics*, vol. 7, no. 2, pp. 124–133, Jun 2002.
- [10] A. B. Schwartz, D. W. Moran, and G. A. Reina, "Differential representation of perception and action in the frontal cortex," *Science*, vol. 303, pp. 380–384, Jan 2004.
- [11] A. Berthoz, *The Brain's sense of movement*, ser. Perspectives in Cognitive Science. London, UK: Harvard University Press, 2000.
- [12] M. Kawato, "Trajectory formation in arm movements: Minimization principles and procedures," in *Advances in Motor Learning and Control*, ser. Human Kinetics, H. N. Zelaznik, Ed. Human Kinetics Publishers, Champaign Illinois, 1996, pp. 225–259.
- [13] A. Hauck, M. Sorg, and T. Schenk, "What can be learned from human reach-to-grasp movements for the design of robotic hand-eye systems?" in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 4, Detroit, Michigan, USA, 1999, pp. 2521–2526.
- [14] M. Bratt, C. Smith, and H. I. Christensen, "Minimum jerk based prediction of user actions for a ball catching task," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. San Diego, Ca, USA: IEEE/RSJ, Oct 2007, pp. 2710–2716.
- [15] C. Smith, M. Bratt, and H. I. Christensen, "Teleoperation for a ballcatching task with significant dynamics," *Neural Networks, Special Issue on Robotics and Neuroscience*, vol. 24, pp. 604–620, May 2008.
- [16] R. Andersson, "Dynamic sensing in a ping-pong playing robot," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 6, pp. 728–739, December 1989.
- [17] H. Hashimoto, F. Ozaki, K. Asano, and K. Osuka, "Development of a pingpong robot system using 7 degrees of freedom direct drive arm," in *1987 International Conference on Industrial Electronics, Control, and Instrumentation*, Nov 1987, pp. 608–615.
- [18] B. Hove and J. Slotine, "Experiments in robotic catching," in *Proceedings of the 1991 American Control Conference*, vol. 1, Boston, MA, Jun 1991, pp. 380–385.
- [19] U. Frese, B. Bauml, S. Haidacher, G. Schreiber, I. Schaefer, M. Hahnle, and G. Hirzinger, "Off-the-shelf vision for a robotic ball catcher," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2001, pp. 1623–1629.
- [20] J. C. Dessing, C. E. Peper, D. Bullock, and P. J. Beek, "How position, velocity, and temporal information combine in the prospective control of catching: Data and model," *Journal of Cognitive Neuroscience*, vol. 17, no. 4, pp. 668–686, 2005.
- [21] M. F. Land and P. McLeod, "From eye movements to actions: How batsmen hit the ball," *Nature Neuroscience*, vol. 3, no. 12, pp. 1340–1345, Dec 2000.
- [22] N. Krüger, "Entwicklung einer humanoiden ball-fang-strategie für ein roboter-hand-arm-system," Master's thesis, Technischen Universität München, 2006.
- [23] P. K. Artemiadis and K. J. Kyriakopoulos, "Teleoperation of a robot arm in 2d catching movements using emg signals and a bio-inspired motion law," in *Proceedings of IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechanics (BioRob)*, Pisa, Italy, Feb 2006, pp. 41–46.
- [24] C. Weber, V. Nitsch, U. Unterhinninghofen, B. Farber, and M. Buss, "Position and force augmentation in a telepresence system and their effects on perceived realism," in *WHC '09: Proceedings of the World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 226–231.
- [25] C. Smith and H. I. Christensen, "Using COTS to construct a high performance robot arm," in *Proceedings of the 2007 IEEE International Conference on Robotics and Automation*. Rome, IT: IEEE, April 2007, pp. 4056–4063.
- [26] S. Hirche and M. Buss, "Human perceived transparency with time delay," *STAR Advances in Telerobotics*, vol. 31, pp. 191–209, 2007.
- [27] M. Bratt, C. Smith, and H. I. Christensen, "Design of a control strategy for teleoperation of a platform with significant dynamics," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Beijing, China: IEEE/RSJ, Oct 2006, pp. 1700–1705.
- [28] K. Kosuge, H. Murayama, and K. Takeo, "Bilateral feedback control of telemanipulators via computer network," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Osaka, Japan, Nov 1996, pp. 1380–1385.
- [29] N. Hogan, "An organizing principle for a class of voluntary movements," *Journal of Neuroscience*, vol. 4, pp. 2745–2754, 1984.
- [30] T. Flash and N. Hogan, "The coordination of arm movements: An experimentally confirmed mathematical model," *The Journal of Neuroscience*, vol. 5, no. 7, pp. 1688–1703, July 1985.
- [31] T. Flash and E. Henis, "Arm trajectory modification during reaching towards visual targets," *Journal of Cognitive Neuroscience*, vol. 3, pp. 220–230, 1991.
- [32] T. Gat-Falik and T. Flash, "The superposition strategy for arm trajectory modification in robotic manipulators," *IEEE Transactions on Systems, Man, And Cybernetics-Part B: Cybernetics*, vol. 29, no. 1, pp. 83–95, Feb 1999.